

Improving Air Traffic Control: Proving New Tools or Approving the Joint Human-Machine System?

Irene Gaillard¹ & Marcel Leroux²

¹Action Recherches Application Matra Irit en Interface Homme Système, France
²Centre d'Etudes de la Navigation Aérienne, France

319-54
p 13

Introduction

From the description of a field problem (i.e., designing decision aids for air traffic controllers), this paper points out how a cognitive engineering approach provides the milestones for the evaluation of future joint human-machine systems.

The European air traffic control (ATC) system has entered a deep crisis: this system is unable to face a tremendous increase of the demand. This is not only the consequence of its inertia since such inertia is normal for any complex system. Short term measures to optimize the present tools appear to be insufficient; these tools have already reached the limits of their evolution capability. A large discussion on how to enhance ATC methods and tools is open. Very ambitious goals are assigned to the future systems, such as the French CAUTRA V project, which plans to double the capacity of the system by the year 2005 and to significantly increase safety. Numerous ambitious projects exist, but none of them has already proved its efficiency or even its feasibility. ATC automation is really short of effective solutions.

Obviously, major technology improvements (FMS, Data Link, 4D-Navigation, computational power) must be extensively used, but in the meantime, full automation cannot be a solution (at least for the next two or three decades); human controllers must remain in the decision making loop. As automation cannot replace human operators, it must assist them. This is a paradox of automation existing within the ATC system: as long as full automation feasibility and efficiency will not be proved (i.e., as long as we will need controllers to make decisions, even in an intermittent way) it is essential to preserve the controllers' skills. No matter the tools that will be designed, it is imperative that human controllers continue to exercise their skills.

Human operators are marked by of flexibility, of capability to deal with unexpected situations, of creativity, and of safety, thanks to their capability to compensate for machine's failures or inadequacies. To preserve these capabilities, we may have to automate "less" than possible from a purely technological point of view.

In the mean time, the human operators are an error factor. From this observation, and for years, system designers thought that the more human operators to be put into a system, the more the risk of error would decrease. In fact, they add another kind of difficulty to the

c-4

N

supervision of the initial system: the difficulty of understanding the behavior of the automatisms that partly monitor the system. Thus, automation makes the operators lose their skills as they know less about the initial system. It creates additional sources of errors and, as reported in numerous examples, the consequences of these errors are much more important than the previous ones. Instead of eliminating human operators with the consequences of depriving the joint system of major benefits and of increasing the risk of errors, it seems more sensible to design a system which is error-tolerant. Such a system cannot be designed only from the technical advances: we must automate in a different way than suggested by the use of technology alone.

The consequences are very important for the design of the future system as well as for its validation. Intuitively, we must not only validate the machine components of the joint system, but we must also verify that the human-machine cooperation works well. As we need to take advantage of the human as a factor of safety and flexibility, we must prove that this requirement is fulfilled. Some aspects of validation can be less critical than with fully automated systems.

Verification and Validation From a Cognitive Engineering Point of View

Cognitive Engineering

Cognitive Engineering as a Method of Designing New Tools. Cognitive engineering has arisen from the expression of a need by numerous system designers: the need to understand what really makes the task difficult for the operators, how this difficulty impairs human performance in order to define the most effective aids (Rasmussen, 1986; De Montmollin & DeKeyser, 1985; Hollnagel 1988), etc. Cognitive engineering is about the multidimensional open worlds which are the effective working context of the operators (Woods & Roth, 1988). Its aim is to understand and describe the present mental activity of operators, given their present tools, and how these mental mechanisms decay under time pressure, fatigue and stress. Cognitive engineering also enables the human factors professional to anticipate how new technologies will modify the activity of operators.

We must not only elicit the knowledge of operators, we must first understand how this knowledge is activated and utilized in the actual problem solving environment. The central question is not to identify the domain knowledge possessed by the practitioner, but rather to point out under which conditions this knowledge is (or is no longer) accessible. This is the problem of defining a validity domain for human performance. Cognitive engineering must also identify and predict the sources of error and the mechanisms of error (Hollnagel, 1991).

When several agents can act on the system, under which conditions can they cooperate efficiently under time pressure? What are the mental resources that are involved and what is the cognitive cost of cooperation (Woods & Roth, 1988)?

Then we have to point out how the present tools are inadequate and determine how the operators compensate for the deficiencies of their tools. Thus, we have to examine how tools provided for an operator are really used by the operators.

Cognitive Engineering as a Method of Validating These Tools. This whole analysis enables us to explain the cognitive model of the operator which is central to defining a global approach to design effective decision aids. In the meantime, this model provides a guideline for validating the joint human-machine system. Validation has no meaning per se; we have to validate according to some criteria which one has to specify. Cognitive engineering enables one to point out the weak aspects of the system as well as those of the human-machine interaction. Thus, it enables the transformation of high level validation requirements into relevant criteria to test the joint human-machine system. It determines which aspects of the machine or of the human-machine interaction must be verified closely so as to guarantee an effective performance of the whole system or to prevent error. Then one can determine or assess the gains along these dimensions.

We must verify that the new joint human-machine system preserves the sources of strong performance and really improves the weak points from a safety point of view as well as from capacity. This suggests that we must assess the performance of the new system with reference to the previous one in real conditions, i.e., whatever the variability of the real world is.

In the meantime, some crucial questions one should always ask are: does the new system preserve the sources of good performance of the operator, or does it preserve the capability of the operator to deal with unanticipated situations or machine deficiencies? These questions become more important as we consider cognitive tools.

Cognitive engineering provides a relevant framework within which one might answer basic questions such as: who has to determine validation criteria, when do we have to experiment so as to verify these criteria, and how and where to experiment? As cognitive engineering is an iterative process at any cycle after designing new tools, the experiments enable one to determine how the global human-machine system evolves, how the bottlenecks in the operator's activity evolve, disappear, decay or are created; what kinds of problems are solved and created by the new system, the new human-machine cooperation philosophy, and what are the consequences of this philosophy on operators' training.

A Field Study: The Case of Air Traffic Controllers

Method

The following is the description of ERATO, a project which is aimed at defining an Electronic Assistant for en-route air traffic controllers. This electronic assistant will include several decision aids. Figure 1 shows that this project includes seven different phases.

At first, we have to elicit the cognitive model of the controllers (1). This model explains the mental mechanisms which are common to all controllers and which enable them to process data and to make real time decision. These mental processes are analyzed for the executive controller, the planning controller and then for both controllers as a whole in order to assess the consequences of cooperation on mental load as well as on global performance. The main goal remains to describe the mental mechanisms involved in decision making process and how these mechanisms evolve and decay under time pressure.

The bottlenecks assessment (2) is a diagnosis phase: we have to point out the sources of poor and good performances of the air traffic controllers given their actual working context. As

long as the situation is not too demanding, controllers can compensate for these bottlenecks; but in very demanding situations these bottlenecks may severely impair the controllers' performance. The assessment of bottlenecks then enables one to specify the basic functions of effective decision aids (3). The specification of the interface (4) also depends on the cognitive model. We need to know in which context these tools will be operated so as to optimize their use.

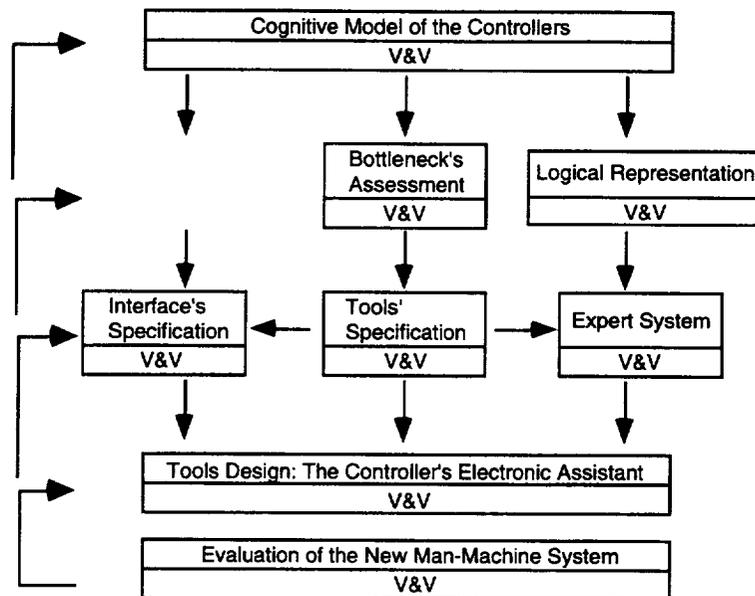


Figure 1. Seven phases for electronic assistant

To define a logical representation of the model (5), we need to combine different laboratory logic so as to build a logical tool adapted to formalize controller's knowledge. The design of the function defined during step (3) involves the use of an expert system which models large subsets of controller's knowledge (6). The expert system provides the two Electronic Assistants with relevant data, so we have to face the problem of the integration of the expert system to real time functions so as to design the two electronic assistants (7).

What Really Makes the Task Difficult

They have to process data which depend on the time factor for:

- Their value
- Their accuracy. When observing air traffic controllers, we found that they spent a lot of time, and a lot of cognitive resources, in eliminating ambiguity. A major reason why

controllers are often unable to make a clear assessment of the situation is based on their representation of predicted time intervals: unless we sink into pure fatalism, we do not anticipate that an event will happen "at" a given time but "about" a given time. The difference is fraught with consequences for the operator.

- Their availability. All the data necessary to make a clear assessment of the situation are not available at a given time; some of them may be definitely unattainable. The operator may have to take decisions in a state of partial ignorance.
- Their flow. The controllers must deal with any cases comprising the dual aspects of both time-dependent information-processing and real-time decision making as: (i) dynamicity vs. inertia of the system or of any subsystem, (ii) lack of data vs. data overflow, and (iii) prolonged low workload vs. tasks overflow. We must also consider the question of how operators can adapt to sudden transitions from any of these aspects to the dual one, for example from a situation of lack of data to a data overflow.
- Data presentation is technology driven and uselessly bulky. Tasks and objectives are not well defined and may severely compete. Risk is so important that the controller has to guard against errors from all the actors (including the machines) in the system.

A Rapid Overview of the Controller's Cognitive Model

The controller anticipates, according to a "normal" routine, behavior of the aircraft in what is called the default world, with reference to the "default logic" that models this kind of reasoning. This default behavior is illustrated by controllers when they use sentences as "normally this aircraft going to Paris Orly will start descent about 30 NM before this fix." The controller does not know the top of descent, but from his experience, he knows that "this will normally happen about here," so he will ignore all potential conflicts that should happen if the given aircraft should start descending earlier to focus all his activity on the most probable conflicts. This is an efficient means of narrowing the range of contingencies to be examined and to increase efficiency: at first all the aircraft are processed as if their behavior always remain consonant with the "normal" behavior.

But to process exceptions, that is to guarantee safety, controllers monitor "sentry parameters." As long as these parameters remain in a "normal" range, all the previous diagnoses or decisions that are inferred from the default world remain valid. But if a sentry parameter drifts outside the expected range, then all the previous plausible inferences have to be revised; some additional conflicts can be created, due to this abnormal behavior. In normal situations, this way of reasoning is an efficient and safe way of making decisions in a state of partial ignorance. But we can observe that, in very demanding situations, the monitoring task may no longer be performed by the controllers. Thus, when outside its validity domain (in too demanding situations), this mechanism may become a major source of errors.

Very often, diagnosis is the result of an ambiguity elimination mechanism. Even when remaining in the default world, the controller is often unable to assess a definite diagnosis. Controllers spend a large amount of time in ambiguity elimination processes. Allowing a doubt is a luxury for the controller; the mastery of doubt is an art. This is performed by associating one (or two) relevant parameter(s) to each undecided situation. To avoid a scattering of resources, these parameters will remain the only ones monitored.

Each conflicting situation, certain or potential, triggers several resolution frames. These frames are a part of the knowledge base common to all controllers. For example, let us consider the following situation: two aircraft converge over the same point, the second climbs through the level of the first. This canonical situation can trigger three resolution frames :

- Clear the climbing aircraft directly to the requested level, under radar monitoring
- Radar vectoring
- Clear the climbing aircraft to a safety level until both have flown over the fix.

A part of the controller's activity is devoted to choose the best frame. Each of these frames may be relatively demanding. In demanding situations, the cognitive cost becomes a basic criteria to choose a frame. Of course while resolving a problem, the controller may have to shift from an inoperative frame to a more relevant one.

According to the assessment of the workload, the controller can instantiate a resolution frame in a more or less efficient way. The controller can also abandon a more elegant frame to shift to a more efficient one: this is the consequence of his or her own resource management policy, according to the problems occurring at a particular time.

All of these mechanisms are a part of the real time data process. This process results in a problem driven organization of the raw data set.

To solve a given conflict, the controller may have to perform actions within far-off time intervals. These time intervals may be very short; if a controller misses the right time span to act on traffic, the very nature and complexity of the problem may change rapidly.

The only way to avoid this is to frequently monitor the position of the relevant aircraft. Obviously this mechanisms is very costly. The controller must shift frequently from one problem to another and, at each shift, must restore the resolution context. When conflicts are complex and under time pressure, this may become a critical task.

Most of the previous tasks can be performed by two controllers, successively or in parallel. Mental mechanisms involved in cooperation are an essential part of the model. Efficient cooperation between the two controllers relies on three factors.

They must have

- The same skills, knowledge and training
- The same representation of effective traffic requirements
- Simultaneously available cognitive resources to exchange information.

When demand increases, these two latter conditions may decay so much that cooperation may no longer be effective. Numerous near misses have been reported due to cooperation failure in too demanding situations. One controller did not even know that some tasks were urgent and important while the other controller thought that these tasks were normally performed. This points out the limits of cooperation based on implicit tasks delegation.

Conclusion: Validity Domain of all the Mental Processes

All these mental mechanisms have a validity domain. We can easily observe how their efficiency decays under stress, time pressure or fatigue.

For example, in demanding situations, the sentry parameters are less monitored and this may lead to errors when an abnormal behavior is not detected soon enough. We can also observe that the assessment of a given situation, including conflict detection and resolution assessment, needs a few tenth of seconds while it can lay on 7 to 8 minutes in very demanding situations; in this case, the controller is confronted with problems associated to numerous shifts from this conflict to concomitant ones, as described before and the risk of error (forgetting a relevant aircraft, choosing the wrong resolution frame, etc.) is high. The validity domain of the mental processes directly depend on the number of aircraft that have to be processed by the controller. This is the reason why we focused on a problem driven presentation of the information.

The efficiency of the mental processes also depends on the capability of the operator to focus his attention on the relevant problem at the right time.

Justification of the Tools

This analysis, which corresponds to phases 3 and 4 of the project, is the key point of the approach. It explains the reasons why each tool has been designed, and the improvements of the joint Human-Machine system that are expected. It also defines the criteria to experiment this system. At this level, there is a deep symbiosis between design and validation; however, this can no longer be true during the experiments.

Information Filtering

Justification. The aim of problem-driven information filtering is to reduce the number of aircraft to be considered at one time. By splitting a too demanding situation in several subsets of aircraft, we can expect that the controllers will have the capability to process these subsets of aircraft very efficiently. As we do not provide the controllers with the results of an automatic conflict detection and resolution, they will have to operate all their mental mechanisms to assess the situation. This should preserve their skills and their capability to deal with any unanticipated situation more properly than now. The expected gain is that, as they will be working on appropriate subsets of aircraft, these mental mechanisms will be much more efficient than now. Thus, we will have to verify that this human-machine cooperation philosophy enhances:

- The way they anticipate in a state of partial ignorance
- The associated sentry-parameters' monitoring processes
- The ambiguity elimination processes: the definite assessment of the situation should be made earlier than now.

- The choice of the relevant resolution frame should be made earlier than now and instanciated in a more "elegant" way.
- The cooperation between controllers should be improved. The information filtering is supposed to enhance the definition of the mental representation. Both controllers' mental representations of the situation should remain consistent over time, as they will be able to update it very easily.

Design Problems. The role allotted to the expert system is to provide the electronic assistants with adequate data in order to show how to organize the raw data in a problem-driven way. Information filtering techniques are under dispute (DeKeyser, 1987). The point is how to make sure that the operator will not need data that is hidden by the system. Data retention and access should not be a source of errors.

In order to answer these precise questions, the expert system must not only encode an exhaustive model of the controller, but we must also carefully define its role in the system.

Description of the Expert System. The first version of the expert system included about 3,000 Prolog first order rules. It processes the same set of data as the controllers have to process now (i.e., the information from the strips) and, when available, the radar information. It includes two main modules. The first computes the default representation of each aircraft. From this representation, the second module associates to each aircraft its relevant environment, called the interfering aircraft subset (IAS).

This environment is composed of :

- The subset of all conflicting aircraft. These conflicts may be certain or potential. This subset is not determined by means of a pure mathematical computation, but rather according to the current expertise of controllers.
- The subset of all the aircraft that may interfere with a "normal" radar resolution of the conflict; that is, all aircraft that may constrain conflict resolution. A normal resolution is a solution which is consistent with the current know-how of controllers.

The relevant environment of an aircraft is typically a problem-driven filtering of information. The IAS represents the relevant working context associated to an aircraft. Such an environment embodies traffic requirements and all information that may be useful to fulfill these requirements. The number of rules is explained by the need to represent current knowledge of the controllers so as to make sure that information filtering really meets the controller's needs.

The definition of the relevant environment is: "according to the traffic requirements, provided the EC works normally, he may need all, or a part of, the displayed data, but he will in no way need any other data."

Discussion. The discussion on the exhaustiveness and the relevance of data filtered by the expert system is central.

- The first answer consists in taking into account the default behavior of the aircraft in a more "prudent" way than the controller. This will result in the display of some aircraft that may be not relevant for the controller. In the most demanding situations (more than 25 aircraft in the sector), the most numerous IAS never include more than 12

aircraft. If one or two additional aircraft are displayed, this is not really a problem. In all cases, the number of aircraft displayed as a result of information filtering remains lower than the maximum efficient processing-capability (about 15 aircraft), while the initial number was largely above this figure.

- The system detects all potentially abnormal behavior of an aircraft in order to advise the controller as soon as possible and to update information filtering accordingly. In future versions, this mechanism should be performed using FMS/Data-Link capabilities.
- But these two first answers do not really solve the problem. The knowledge elicited in the expert system defines a set of "normal behaviors" of the controllers. But whatever the number of rules can be, it is impossible to represent the whole knowledge of all the controllers. Should we be able to do this, we should have to deal with controllers' errors or creativity. The solution defined in Erato consists of considering the expert system as a default representation of the controllers. To guard against the consequences of human error or creativity, (i.e., unexpected behavior) a monitoring process exists. This process is inspired by the natural sentry-parameters' monitoring process of the controllers. This monitoring process will detect any discrepancy between the actual position of all aircraft and any of the possible position as it could result from a "normal" behavior of the controller. When necessary, this process will trigger an alarm to advise the controller that the previous information filtering is no longer relevant and has been updated. We have to make sure that this mechanism is efficient in demanding situations, and that the controller is not overwhelmed by the warnings. In other terms, the expert system must be accurate enough.

This monitoring process associated with the expert system allows the electronic assistant to smoothly adapt to operator error and creativity. Such information filtering is error tolerant.

These results are used by several functions of the electronic assistant: simulation of problem's resolution, extrapolation, memorization aid, data transfer from one device to the other, cognitive-resources-management aid. The problem-driven information filtering allows the controller to concentrate on well-formulated problems in order to operate in a more efficient and creative way. This function substitutes a set of easily manageable problems for the initial complex situation. The basic information filtering will be used by all the functions of the Electronic Assistant.

The Extrapolation Function

Justification. This function will enable both controllers to improve problem formulation. As long as the aircraft are not in radar contact, the controller has to process data from the strips. In demanding situations, it is easy to observe that the strips are not read in an exhaustive way, which can cause severe errors. This function will display the situation at any future time, taking into account the default behavior of all the aircraft. It substitutes a graphical display for an alphanumeric representation of data, and this will improve the choice of the right resolution frames. It will also enable the controller to more easily assess where uncertainty lies, and consequently to more efficiently point out the parameters which are relevant to eliminate ambiguity.

Design Problems. It is commonly admitted that operators spend a significant part of their activity in compensating for tool deficiencies. An ill-adapted interface can significantly devalue the results of information filtering. We have suggested that very often the reference used by the controller is not a temporal one but a spatial one: the question "when will you act on this aircraft" is answered "There." Thus, the interface will enable the controller to drag an aircraft along its trajectory with the mouse; all the other aircraft will move accordingly. This interface responds in the way the controller anticipates. In some situations, the controller refers to a temporal reference. The interface will display simultaneously the time corresponding to the simulated position of the aircraft. However, should this interface have had only one of these references, the controller should have had to mentally convert distances into time intervals. In demanding situations this could represent a significant additional workload.

Simulation Functions. These functions will allow the controller to experiment with different resolution frames and to answer questions such as "what would happen if I climb this aircraft to..." or "Is it better to set course directly to..." The expert system will deliver the simulated information filtering. These answers will be updated until the controller has made a decision. For the time being, the controllers have no tool that assists them in performing this task.

Memorization Aids. The controller will have the ability to indicate the trajectory section where he intends to vector an aircraft. When the aircraft flies over this point (or abeam), a warning will be triggered; then, after having consulted the filtered information, the controller will initiate his decision. This should solve both problems of keeping onto memory the "right time to act" and context-resolution real-time updating.

Data Transfer Between Electronic Assistants

This function is aimed at improving cooperation between controllers. It enables them to transfer any filtered information or simulated result from one position to the other. Using this small set of data, the two controllers will have the same mental representation of the requirements of the situation. The capability to save incoming messages into a letter box until the controller has time to process them should help to solve the problem of simultaneous availability of both controllers to exchange information.

The Reminder

Justification. We observed how it may become difficult for the controller to focus complete attention on the right problem at the right time. The reminder consists of a specific window of the electronic assistant where each problem will be associated a label. A problem is defined as a conflicting situation involving two or more aircraft. The labels are positioned according to the urgency, and the display of the relative urgency of problems should enable the controller to avoid wasting cognitive resources on non-urgent and unimportant tasks while the short term situation decays. In normal operations, this should allow the controller to objectively manage all cognitive resources.

The aim of the reminder is to show what the traffic requirements are and their urgency to the two controllers. There are several ways to split a given situation into relevant problems. This variability can be observed for several different controllers as well as for any given controller,

according to his cognitive resources management philosophy. The more the situation is felt to be demanding, the more the controller will split it in "little" problems and solve these problems in a very tactical way: with short term solutions. If the controller feels that the situation is mastered he or she will consider these elementary problems as a part of a whole and solve them in a more strategic way. Statistically, about a quarter of the problems may be more or less broken down, while the three other quarters are always described in the same way by controllers.

Design Problems. The reminder will propose labels by default. Most of these labels will correspond to the effective needs of the controller; some will not. Thus, the controller will have the capability of editing these labels.

When a situation can be described as a single problem or as several problems, the reminder will propose the simplest situation for three reasons:

- It corresponds to the need of the controller when the situation is the most demanding. If the controller wants to concatenate some sub-problems, the situation is probably not too demanding, and there is time to do this properly. This is a means of avoiding clumsy automation: the interface assists the operator more in the most demanding situations.
- It is easier to concatenate the labels than to split them.
- This enables the system to point out more information on each sub-problem.

Validation Techniques

Classically, we define dependability as that property of a computing system which allows reliance to be justifiably placed on the service it delivers (Laprie, 1987). We can point out four classical methods regarding dependability-procurement or dependability-validation: fault avoidance, fault tolerance, fault removal and fault forecasting. These definitions can be applied to a complex heterogeneous human-machine system as the ATC system as well as to any of its machine components. In the first case, the users are the airlines (or their passengers), while in the second case the user is defined as the controller or any subsystem.

The specification of decision aids relies on a philosophy of future human-machine cooperation, whether this philosophy is clearly defined or not. The central question is to make sure that this cooperation fulfills the initial requirements regarding capacity and safety. To answer this question, we have to choose the right parameters to be evaluated and then determine the minimum set of experiments to get a significant amount of data (Woods & Sarter, 1992). Some basic questions must be answered about the robustness of the joint human-machine system: is it error tolerant (Foster, 1992)? Does its organization allow a quick and easy correction of errors (Reason, 1992)?

The design of decision aids implies an analysis, either implicit or explicit, of operator deficiencies and of the most effective means to compensate for these deficiencies. The ultimate step of the verification and validation process should be the verification of these initial assumptions (Hopkin, 1992).

The validation of the Electronic Assistant is threefold (Leroux, 1992).

- The cognitive model has been verified through the analysis of the behavior of air traffic controllers during simulations of demanding traffic (Leroux, 1991). The observation of the controllers and their comments enabled us to point out the mental processes and the bottlenecks as described in the model.
- The philosophy of human-machine cooperation and the specification of the electronic assistants have already been presented to more than 180 fully qualified controllers.
- The expert system is used to provide the various functions of the electronic assistant with adequate data. To validate this knowledge-based module, we have to check that the outputs are acceptable by controllers on real conditions; i.e., that they always include at least the minimum set of information. Although the validation of the knowledge based module has not been carried out, it has been successfully confronted to the most usual problems. However, we will have to validate very carefully the monitoring process, as it guarantees the relevance of the outputs. We must prove that this mechanism detects all discrepancies between the observed state of the world and the expected one. The trust of the operator on the machine relies on the efficiency of this mechanism. Thus, the validation of the knowledge-based module will be twofold: from a pure dependability point of view and from a human factors point of view. This will be performed by two different teams with different techniques to experiment.
- The first level of validation of the Electronic Assistant will consist of:
 - Experimenting the interface of each function
 - Verifying that it really improves the target bottlenecks
 - And in making sure that it does not decay some sources of good performance. The criteria for experimenting with these functions are directly issued from the cognitive model, as previously described.

Then we will have to assess the validity domain of each function: are they really efficient for situations where the controller needs an effective aid?

- After that, we will have to validate the electronic assistant as a whole in order to analyze if the function-as-a-part-of-a-whole loses properties or acquires unexpected ones.
- Finally, we must assess the performance of the joint human-machine system and answer questions such as:
 - How is this cooperation philosophy accepted by controllers?
 - How will it modify the activity of the controllers?
 - Does it enable them to work in a more efficient and creative way?
 - Does it provoke a loss of vigilance or of skill?
 - Does it improve the global performance, from capacity and safety point of view?
 - What are the consequences on training?
 - Does it enable a progressive and "soft" integration of technological advances in avionics?

References

- Bainbridge, L. (1987). Ironies of automation. In J. Rasmussen, K. Duncan, & J. Leplat (Eds.), *New technology and human error*. John Wiley & Sons.
- Celio, J. C. (1990, February 2). Controller perspective of AERA (MP-88W00015 rev1 / Project 1764c Dept W42). Mitre Corporation.
- De Keyser, V. (1987). How can computer-based visual displays aid operators? *International Journal of Man-Machine Studies*, 27, 471-478.
- De Montmollin, M., & De Keyser, V. (1985). Expert logic vs. operator logic. In G. Johannsen, C. Mancini, & L. Martensson (Eds.), *Analysis, design, and evaluation of man-machine systems*. CEC-JRC Ispra, Italy: IFAC.
- Foster, H. D. (1992). Resilience theory and system valuation. In J. A. Wise, V. D. Hopkin, & P. Stager (Eds.), *Verification and Validation of Complex and Integrated Human-Machine Systems*. Vimeiro, Portugal: NATO Advanced Study Institute.
- Hollnagel, E. (1988). Information and reasoning in intelligent decision support systems. In Hollnagel, Mancini, & Woods (Eds.), *Cognitive Engineering in Complex Dynamic Worlds*. London: Academic Press.
- Hollnagel, E. (1991). The phenotype of erroneous actions: Implications for HCI design. In A. Weir (Ed.), *Human-Computer Interaction and Complex Systems*. London: Academic Press, Computers and people series.
- Hopkin, V. D. (1992). Verification and validation: Concept issues and applications. In J. A. Wise, V. D. Hopkin, & P. Stager (Eds.), *Verification and Validation of Complex and Integrated Human-Machine Systems*. Vimeiro, Portugal: NATO Advanced Study Institute.
- Laprie, J. C. (1987). Dependable computing and fault tolerance at LAAS: a summary. In A. Avizienis, H. Kopetz, & J. C. Laprie (Eds.). New York: Springer-Verlag Wien.
- Leroux, M. (1991a). Premier bilan des experimentations du modele Erato, specification d'outils cooperatifs pour le controleur. Rapport CENA 91538.
- Leroux, M. (1991b). Erato (en route air traffic organizer). *Proceedings of the Sixth International Symposium on Aviation Psychology*, Columbus (Ohio).
- Leroux, M. (1992). The role of verification and validation in the design process of knowledge based components of air traffic control systems. In J. A. Wise, V. D. Hopkin, & P. Stager (Eds.), *Verification and Validation of Complex and Integrated Human-Machine Systems*. Vimeiro, Portugal: NATO Advanced Study Institute.
- Rasmussen, J. (1986). *Information processing and human-machine interaction: An approach to cognitive engineering*. New York: North Holland.
- Reason, J. (1992). The identification of latent organizational failures in complex systems. In J. A. Wise, V. D. Hopkin, & P. Stager (Eds.), *Verification and Validation of Complex and Integrated Human-Machine Systems*. Vimeiro, Portugal: NATO Advanced Study Institute.
- Roth, E. M., Bennett, K. B. & Woods, D. D. (1987). Human interaction with an "intelligent" machine. *International Journal Of Man-machine Studies* 27, 479-525
- Villiers, J. (1968). Essai sur l'evolution du controle de la circulation arienne en route.
- Villiers, J. (1992). L'homme face aux systemes techniques complexes. *Futuribles*, 167, 111-125.
- Woods, D. D. (1986). Paradigms for intelligent decision support. In Hollnagel, Mancini, & Woods (Eds.), *Intelligent decision support in process environments*. New York: Springer-Verlag.
- Woods, D. D., & Sarter, N. B. (1992). Field experiments for assessing the impact of new technology on human performance. In J. A. Wise, V. D. Hopkin, & P. Stager (Eds.), *Verification and Validation of Complex and Integrated Human-Machine Systems*. Vimeiro, Portugal: NATO Advanced Study Institute.

